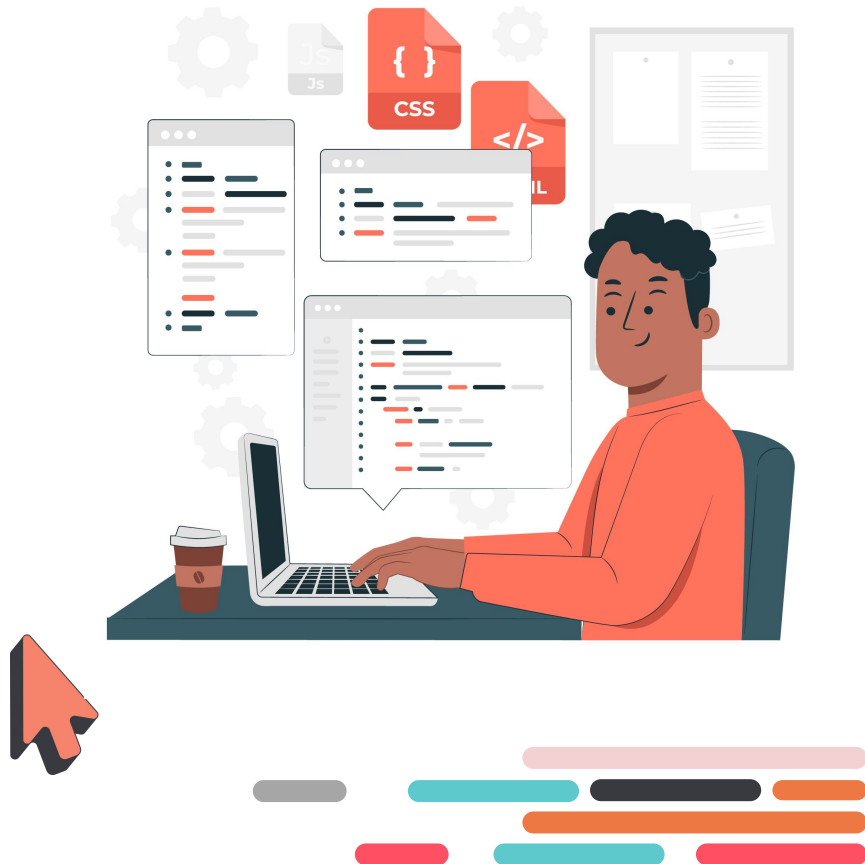




KEMNAKER



JWT Auth





KEMNAKER

sanbercode



Outline

- JWT
- Penerapan JWT Auth di Express.js





JWT





Apa itu JWT?

JWT, atau JSON Web Token, adalah standar terbuka (RFC 7519) yang mendefinisikan format token yang ringan dan dapat dibaca oleh mesin. JWT biasanya digunakan untuk mentransfer klaim antara pihak-pihak dalam cara yang dapat diverifikasi. Informasi ini dapat diverifikasi dan dipercayai karena ditandatangani secara digital. JWT dapat ditandatangani menggunakan algoritma HMAC (Hash-Based Message Authentication Code) atau menggunakan pasangan kunci kriptografis (RSA atau ECDSA).

Sebuah token JWT terdiri dari tiga bagian terenkripsi dalam bentuk Base64 URL, yang masing-masing disebut sebagai header, payload, dan signature

<https://jwt.io/>





Penerapan JWT Auth pada Express

untuk menerapkan JWT Auth pada Express install library dibawah ini:

```
npm install dotenv jsonwebtoken bcrypt
```

- **dotenv** digunakan untuk membaca environment variable dari file `.env`
- **jsonwebtoken** untuk menggunakan fungsi-fungsi JWT
- **bcrypt** untuk enkripsi





KEMNAKER



Membuat Register

```
app.post('/register', async (req, res) => {  
  const { username, password } = req.body;  
  // Check if the user already exists  
  if (pengecekan existing user) {  
    return res.status(409).json({ message: 'User already exists' });  
  } // Hash the password  
  
  const hashedPassword = await bcrypt.hash(password, 10);  
  // kode save user disini  
  // Add the new user to the users array (or your database  
  res.status(201).json({ message: 'User registered successfully' });  
});
```





KEMNAKER



Membuat Login

```
app.post('/login', async (req, res) => {  
  const { username, password } = req.body;  
  const user = dapatkan user dari database;  
  if (!user || !(await bcrypt.compare(password, user.password))) {  
    return res.status(401).json({ message: 'Invalid credentials' });  
  }  
  const accessToken = jwt.sign({ username: user.username, id: user.id },  
process.env.JWT_SECRET);  
  res.json({ accessToken });  
});
```





KEMNAKER



Membuat Middleware Jwt Auth

tambahkan juga variabel JWT_SECRET pada file .env

```
const authenticateJWT = (req, res, next) => {  
  
  const authHeader = req.header('Authorization');  
  if (!authHeader) return res.status(401).json({info: "Unauthorized"});  
  const token = authHeader.split(" ")[1]  
  jwt.verify(token, process.env.JWT_SECRET, (err, user) => {  
    if (err) return res.status(401).json({info: "Unauthorized"});  
    req.user = user;  
    next();  
  });  
};
```





KEMNAKER

sanbercode



Kesimpulan

- JWT
- Penerapan JWT Auth di Express.js





Terima Kasih

